



Title: Android application for tracking garbage collection vehicle in Huauchinango Puebla

Authors: BARRON-CASTILLO, Jorge Alfredo, HERNÁNDEZ-LUNA, Aldo, TORRES-JIMÉNEZ,
Jacinto and LUNA-CARRASCO, Claudia Yadira

Editorial label ECORFAN: 607-8695
BCIERMMI Control Number: 2022-01
BCIERMMI Classification (2022): 261022-0001

Pages: 12
RNA: 03-2010-032610115700-14

ECORFAN-México, S.C.
143 – 50 Itzopan Street
La Florida, Ecatepec Municipality
Mexico State, 55120 Zipcode
Phone: +52 1 55 6159 2296
Skype: ecorfan-mexico.s.c.
E-mail: contacto@ecorfan.org
Facebook: ECORFAN-México S. C.
Twitter: @EcorfanC

www.ecorfan.org

Holdings		
Mexico	Colombia	Guatemala
Bolivia	Cameroon	Democratic
Spain	El Salvador	Republic
Ecuador	Taiwan	of Congo
Peru	Paraguay	Nicaragua

Introducción

En el municipio de Huauchinango puebla se tiene la problemática que la gente no tiene certeza del momento en que pasara el carro recolector de basura, es por eso que al no saber el momento correcto para sacar su basura a la calle, está la saca desde la mañana ya sea porque tiene que trabajar, o salir de su domicilio por cualquier situación.

Dada esta situación, la gente deja su basura en plena vía pública esperando que pase el carro recolector, esto genera puntos de infección, acumulación de animales como perros, ratas etc., que también son portadores de enfermedades, además de dar un mal aspecto a la ciudad y obstruir los pasos peatonales.

Metodología

De acuerdo a la experiencia laboral obtenida durante años pasados en el desarrollo de aplicaciones de escritorio. La metodología seleccionada será “programación extrema (XP)”, esta metodología siempre fue utilizada durante los desarrollos dando un resultado deseado en las implementaciones ya que su estructura nos permite establecer tiempos de entrega y agilizar procesos gracias a sus iteraciones cortas.

PROGRAMACIÓN EXTREMA (XP)

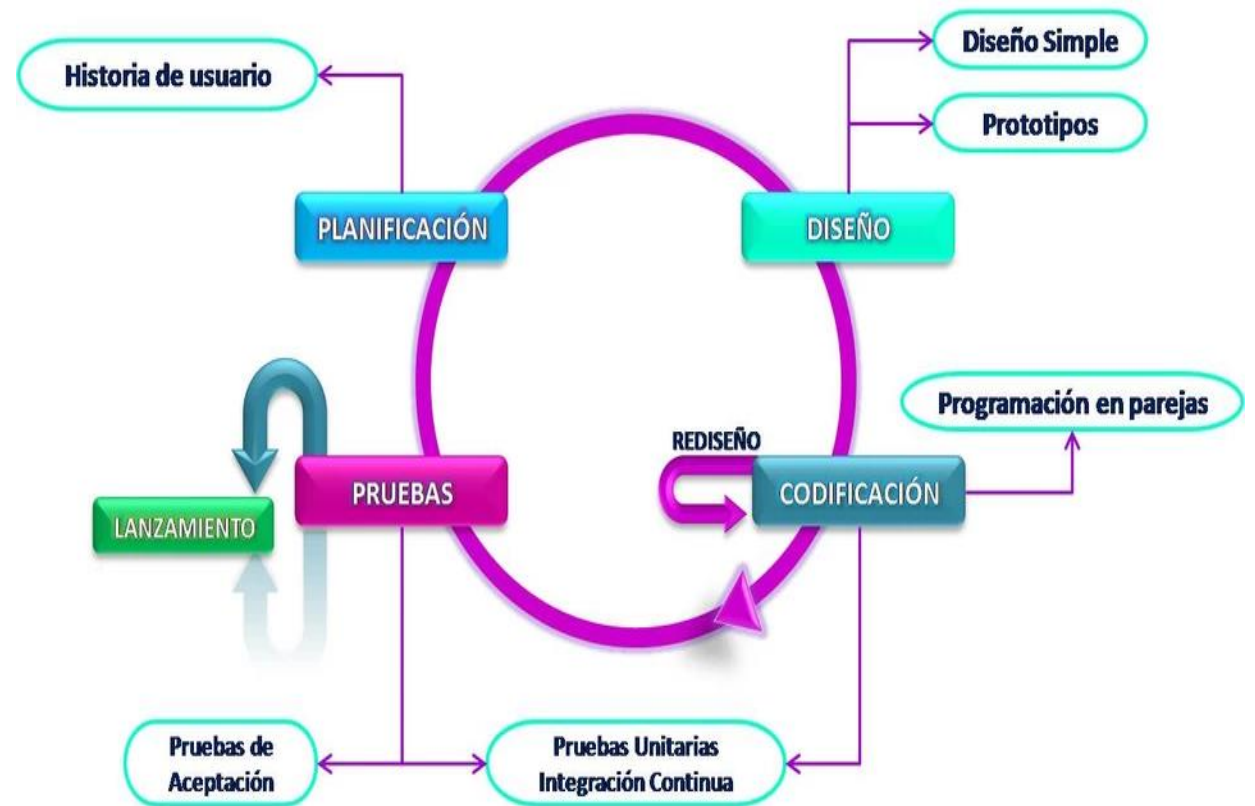


Figura 1 Metodología.

Metodología - Análisis

Se hizo un análisis inicial, en el cual se comparaban los tipos de tecnologías a utilizar para la realización del proyecto, comparando costos, tamaño del dispositivo, y características.

se opto fue el desarrollo de una aplicación móvil. Aprovechando que los equipos celulares ya cuentan con su propio rastreador GPS y son programables. Haciendo esto más adaptable al proyecto.

Para el desarrollo de la aplicación móvil de ha seleccionado que la primera versión deberá ser para dispositivos Android. Ya que en la ciudad de Huauchinango aproximadamente el 80% de personas usa esta plataforma.

Su desarrollo será en Android Studio ya que este IDE es el nativo para desarrollo de aplicaciones Android, utilizando lenguaje JAVA, con API de Google maps para poder tener una geolocalización más exacta del dispositivo.

	Dispositivo Android	Rastreador Gps	Arduino
Rastreo satelital	X	X	X/con modulo
Compatible con red GSM	X	X	X/con modulo
Fuente de alimentación	Batería recargable	Batería recargable	USB o eliminador
Tiempo de uso	24 Hrs. Aprox.	4 a 6 días	---
Costo	Depende del equipo	\$400	\$250

Tabla 1 Comparación tecnologías propuestas.

Metodología - Desarrollo

Para poder utilizar el API de google se requiere de una llave, la cual se configura desde Google Cloud Platform, para esto debemos dirigirnos al menú desplegable y seleccionar API y servicios Una vez seleccionada la sección de API y servicios se debe seleccionar la opción de biblioteca y enseguida se deben habilitar los servicios de MAPS SDK, PLACES API y MAPS Java Script API Despues de haber habilitado los servicios de google se debe crear una clave de API, la cual es necesaria para poder utilizar los servicios de google, para esto se debe seleccionar la sección de credenciales y a continuación seleccionar crear credenciales

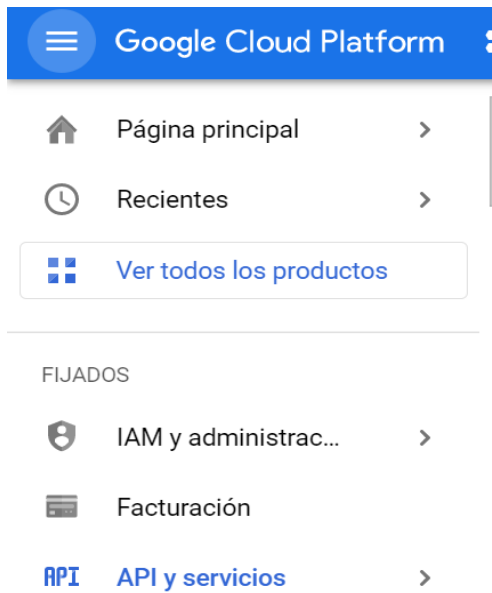


Figura 2 Menú desplegable, selección de API

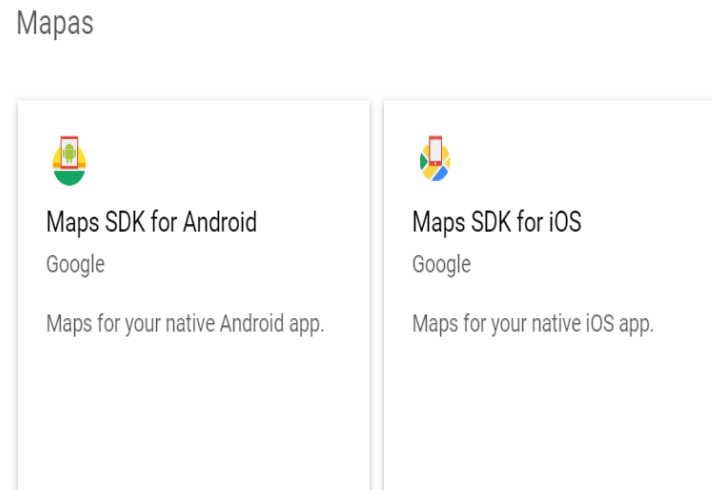


Figura 3 Servicios de google

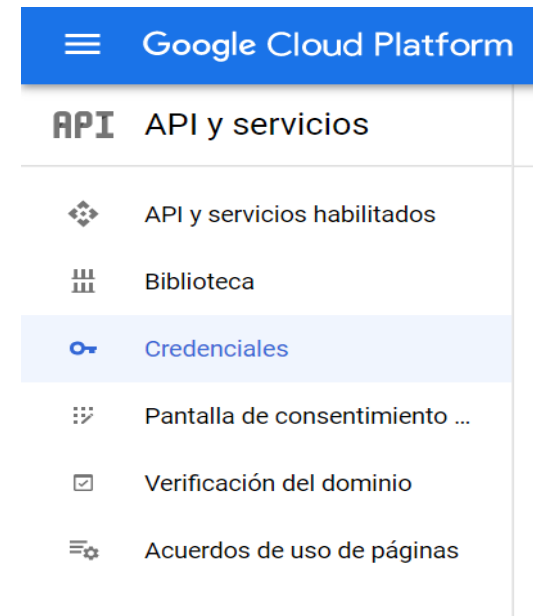


Figura 4 Creación de credenciales de API.

Metodología - Desarrollo

Para poder hacer uso de mapas de google. Se debe usar la llave que se generó durante la configuración de las API's, esta llave debe de usarse en el manifiesto de Google, así como los permisos necesarios para que la aplicación pueda usar la geolocalización del dispositivo y el internet Una vez terminado el manifiesto, se desarrolla la pantalla, en este caso la aplicación deberá mostrar la ubicación real del dispositivo al momento de presionar un botón, el que llamamos coordenadas, a continuación, se muestra el diseño de la primera versión de esta aplicación.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Figura 5 Programación de manifiesto de Android.

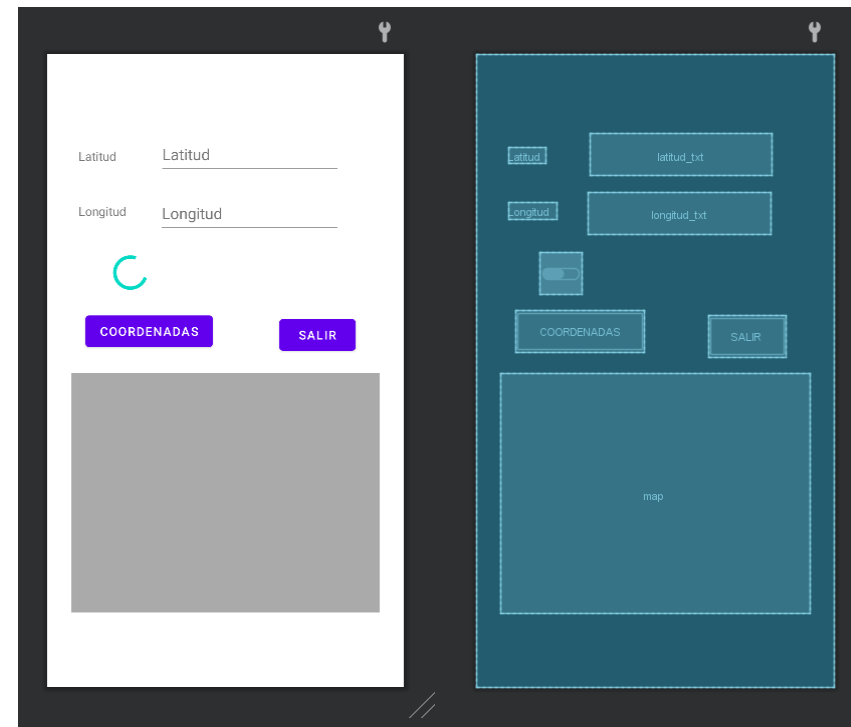


Figura 6 Diseño inicial de aplicación Android.

Metodología - Desarrollo

Se deben crear variables de tipo objeto que se encuentra en la pantalla de diseño, esto para hacer una conexión entre el código de nuestra aplicación y el diseño de la misma, así como las variables globales necesarias para el desarrollo de nuestro programa.

En el método onCreate se hará la conexión de los objetos y el diseño de la aplicación, seguido se crea el método ObtenerCoordenadasActual, en el cual se revisan si la aplicación ya cuenta con los permisos necesarios para la ejecución del programa. En caso de no tener permisos aun le pregunta al usuario si concede los permisos del uso de internet y geolocalización

```
public class MainActivity extends AppCompatActivity {
    public static final int REQUEST_CODE = 1;
    EditText lat,lon,dir;
    Button obtener,salir;
    ProgressBar progress;
    FusedLocationProviderClient fusedLocationProviderClient;
    MapView mapa;

    double latitud, longitud;
}
```

Figura 7 Creación de variables globales.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    lat= (EditText) findViewById(R.id.latitud_txt);
    lon = (EditText) findViewById(R.id.longitud_txt);

    obtener = (Button) findViewById(R.id.Obtener_btn);
    salir = (Button) findViewById(R.id.salir_btn);

    progress = (ProgressBar) findViewById(R.id.progressBar);
    mapa = (MapView) findViewById(R.id.map);
}

public void ObtenerCoordenadasActual(View view) {
    if (ContextCompat.checkSelfPermission(getApplicationContext(),
        Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED){
        ActivityCompat.requestPermissions( activity: MainActivity.this,
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},REQUEST_CODE);
    } else{
        GetCoordenada();
    }
}
```

Figura 8 Inicialización de variables y permisos.

Metodología - Desarrollo

Por último, se crea el método principal, que va a ser el que usa el botón de nuestra aplicación para obtener las coordenadas actuales del dispositivo. Este también revisa que se tengan los permisos necesarios para poder hacer uso de las clases `locationRequest` que nos permite hacer el uso de la geolocalización de nuestro dispositivo. Y mostrar las coordenadas actuales en la pantalla de nuestra aplicación

Figura 9 Método principal.

```
private void GetCoordenada(){
    try {
        progress.setVisibility(View.VISIBLE);

        LocationRequest locationrequest = new LocationRequest();
        locationrequest.setInterval(10000);
        locationrequest.setFastestInterval(3000);
        locationrequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

        if (ActivityCompat.checkSelfPermission( context: this,
        Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED){
            return;
        }

        LocationServices.getFusedLocationProviderClient( activity: this).requestLocationUpdates(locationrequest,
        new LocationCallback() {
            @Override
            public void onLocationResult(@NonNull LocationResult locationResult) {
                super.onLocationResult(locationResult);
                LocationServices.getFusedLocationProviderClient( activity: MainActivity.this).removeLocationUpdates( locationCallback: this);
                if(locationResult != null && locationResult.getLocations().size()>0){
                    int latestLocationIndex = locationResult.getLocations().size()-1;
                    latitud = locationResult.getLocations().get(latestLocationIndex).getLatitude();
                    longitud = locationResult.getLocations().get(latestLocationIndex).getLongitude();
                    lat.setText(String.valueOf(latitud));
                    lon.setText(String.valueOf(longitud));
                }
            }
        }, Looper.myLooper());};catch (Exception ex){
        System.out.println("Error es: " + ex);
    }
}
```


Resultados

La aplicación fue probada para saber la certeza de las coordenadas actuales, una vez presionado el botón de coordenadas nos arroja las coordenadas actuales de nuestro dispositivo y Las coordenadas fueron revisadas desde Google maps.

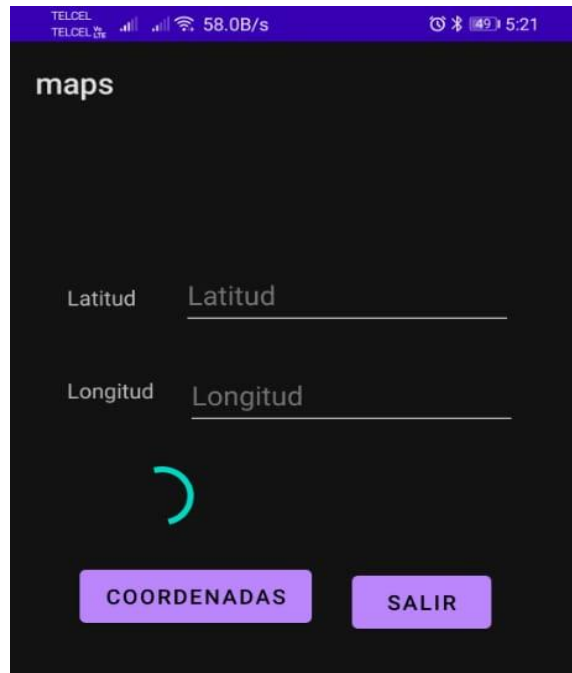


Figura 10 Pantalla principal.

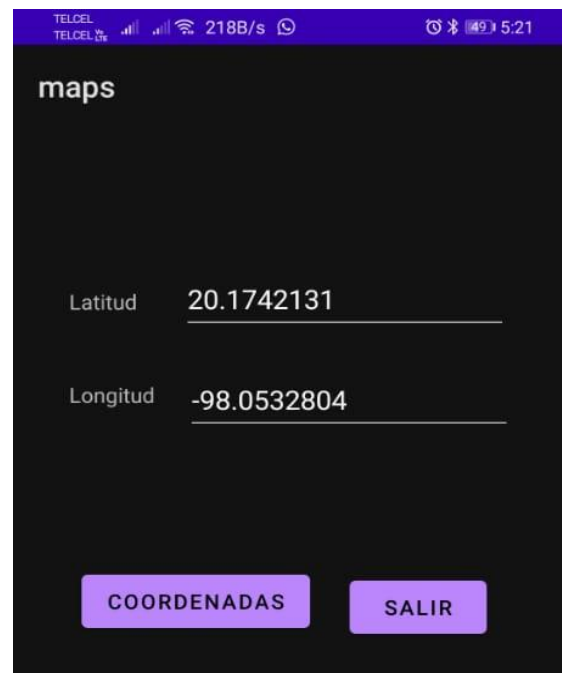


Figura 11 Ejecución de programa.

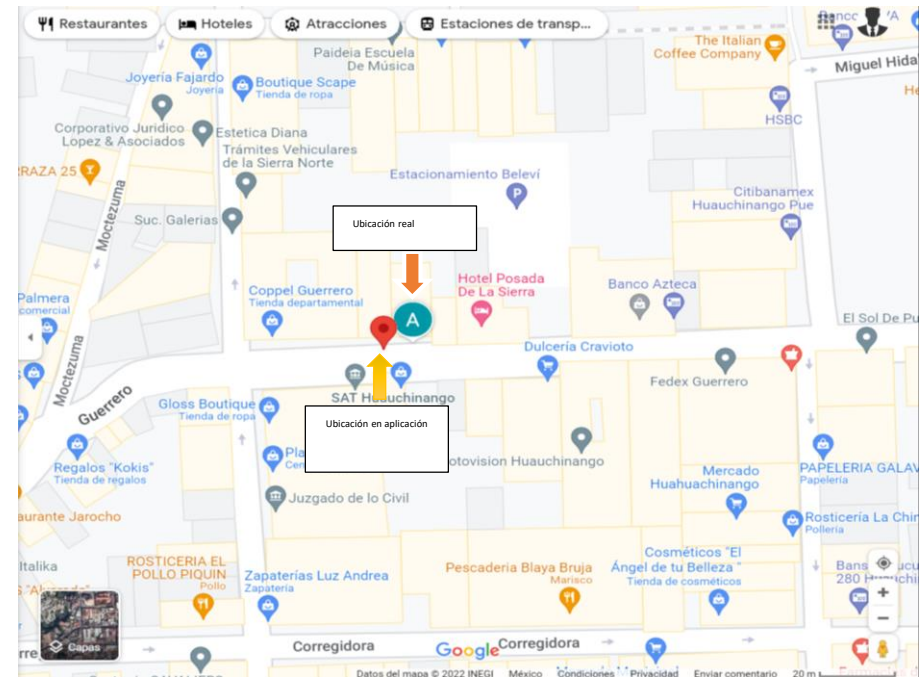


Figura 12 Vista de ubicación real.

Resultados

Los resultados de la aplicación son los esperados, ya que nos muestra la ubicación de manera rápida y bastante precisa, con un margen de diferencia aproximadamente 5 metros esto no afecta el resultado del propósito de la aplicación, lo cual nos indica que es la tecnología más viable para la ejecución del proyecto utilizando una aplicación que deberá llevar el equipo Android del chofer del vehículo recolector de basura.

Para la siguiente fase, esta aplicación mandará cíclicamente a un servidor su ubicación actual, para que entonces este vaya almacenando y así poder tener una bitácora, a la cual también existirá una aplicación cliente, que será la que muestre el mapa obteniendo la última ubicación almacenada en el servidor de manera constante para dar un seguimiento en tiempo real de el vehículo recolector. Así como mostrar los avisos necesarios para que el usuario saque la basura en el momento adecuado o en caso de ser necesario no sacar su basura.

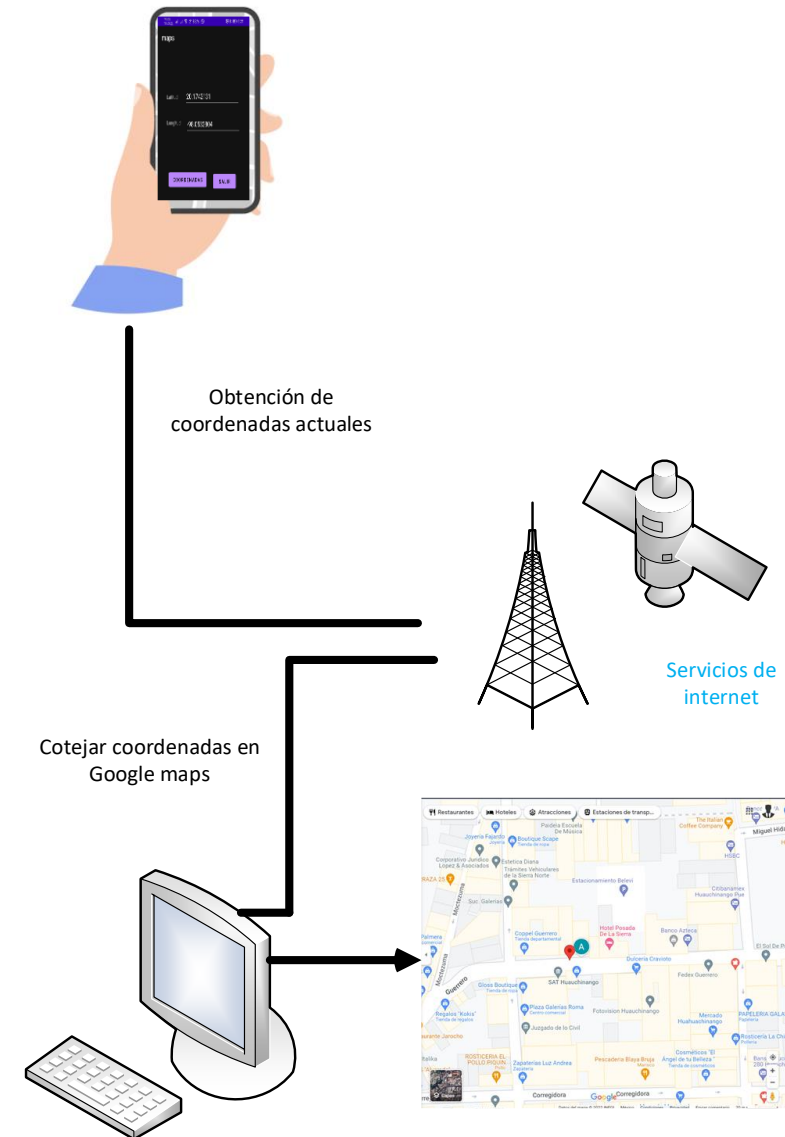


Figura 13 Diagrama general.

Conclusiones

Esta prueba inicial nos da una certeza de que la tecnología más adecuada para el proyecto de seguimiento al vehículo de basura será utilizar el dispositivo Android que lleve el vehículo. Ya que su tiempo de respuesta y su precisión son las adecuadas, agregando, que siendo programable el dispositivo se pueden añadir mejores funciones en versiones futuras las cuales sean útiles dependiendo de la necesidad de los usuarios.

Referencias

Cale Pillco, D. A. (2019). Sistema de rastreo vehicular mediante GPS implementando tecnología JAVA en Limón Indanza. *Revista Tecnológica Amazónica*, 2(1), 33–36. <https://itsfo.edu.ec/web/wp-content/uploads/2022/03/Revista-2019.pdf>

Guzmán Acán, L. G. (2018), Evaluación y diseño de un sistema de rastreo satelital para el monitoreo y control de las rutas terrestres asignadas diariamente en tiempo real (Proyecto de investigación y desarrollo para optar el título en magister en sistemas de telecomunicaciones). Escuela Superior Politécnica de Chimborazo.

<http://dspace.esPOCH.edu.ec/handle/123456789/8731>

Luna Santos, R. (2022), Prototipo de dispositivo electrónico para envío de ubicación, temperatura y ritmo cardiaco de una persona a un servidor de internet para el monitoreo desde una app móvil (Tesis para obtener el grado de maestro en tecnologías de la información). Instituto Tecnológico superior de Huauchinango.

DOI: <https://doi.org/10.36825/RITI.09.18.007>

Referencias

Henao Melo, L. G. (2014), Diseño de un equipo de rastreo satelital de elementos usando tecnologías GPS y GSM (master Thesis) Universidad Tecnológica de Pereira.

<https://core.ac.uk/download/pdf/71398213.pdf>

Tapuy Cerda, W. L (2019), Diseño de un prototipo electrónico para el encendido de un vehículo Renault Logan, mediante tecnología NFC y rastreo satelital GPS (Tesis para obtener el grado de ingeniero en electrónica, redes y telecomunicaciones). Escuela superior politécnica de Chimborazo.

<http://dspace.esPOCH.edu.ec/handle/123456789/10961>

Google Cloud Platform. (s. f.). Google Cloud Platform. Recuperado 6 de junio de 2022



ECORFAN®

© ECORFAN-Mexico, S.C.

No part of this document covered by the Federal Copyright Law may be reproduced, transmitted or used in any form or medium, whether graphic, electronic or mechanical, including but not limited to the following: Citations in articles and comments Bibliographical, compilation of radio or electronic journalistic data. For the effects of articles 13, 162,163 fraction I, 164 fraction I, 168, 169,209 fraction III and other relative of the Federal Law of Copyright. Violations: Be forced to prosecute under Mexican copyright law. The use of general descriptive names, registered names, trademarks, in this publication do not imply, uniformly in the absence of a specific statement, that such names are exempt from the relevant protector in laws and regulations of Mexico and therefore free for General use of the international scientific community. BCIERMMI is part of the media of ECORFAN-Mexico, S.C., E: 94-443.F: 008- (www.ecorfan.org/booklets)